Bridging Concept Identification for Constructing Information Networks from Text Documents

Matjaž Juršič[1], Borut Sluban[1], Bojan Cestnik[1, 2], Miha Grčar[1], and Nada Lavrač[1, 3]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] Temida d.o.o., Ljubljana, Slovenia
[3] University of Nova Gorica, Nova Gorica, Slovenia
{matjaz.jursic,borut.sluban,bojan.cestnik,
miha.grcar,nada.lavrac}@ijs.si

Abstract. A major challenge for next generation data mining systems is creative knowledge discovery from diverse and distributed data sources. In this task an important challenge is information fusion of diverse mainly unstructured representations into a unique knowledge format. This chapter focuses on merging information available in text documents into an information network – a graph representation of knowledge. The problem addressed is how to efficiently and effectively produce an information network from large text corpora from at least two diverse, seemingly unrelated, domains. The goal is to produce a network that has the highest potential for providing yet unexplored cross-domain links which could lead to new scientific discoveries. The focus of this work is better identification of important domain-bridging concepts that are promoted as core nodes around which the rest of the network is formed. The evaluation is performed by repeating a discovery made on medical articles in the migraine-magnesium domain.

Keywords: Knowledge Discovery, Text Mining, Bridging Concept Identification, Information Networks, PubMed, Migraine, Magnesium.

1 Introduction

Information fusion can be defined as the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human and automated decision making [5]. Creative knowledge discovery can only be performed on the basis of a sufficiently large and sufficiently diverse underlying corpus of information. The larger the corpus, the more likely it is to contain interesting, still unexplored relationships.

The diversity of data and knowledge sources demands a solution that is able to represent and process highly heterogeneous information in a uniform way. This means that unstructured, semi-structured and highly structured content needs to be integrated. Information fusion approaches are diverse and domain dependent. For instance, there are recent investigations [7, 19] in using information fusion to support

M.R. Berthold (Ed.): Bisociative Knowledge Discovery, LNAI 7250, pp. 66–90, 2012.

scientific decision making within bioinformatics. Smirnov et al. [22] exploit the idea of formulating an ontology-based model of the problem to be solved by the user and interpreting it as a constraint satisfaction problem taking into account information from a dynamic environment.

In this chapter we explore a graph-theoretic approach [1, 2] which appears to provide the best framework to accommodate the two dimensions of information source complexity – type diversity as well as volume size. Efficient management and processing of very large graph structures can be realized in distributed computing environments, such as grids, peer-to-peer networks or service-oriented architectures on the basis of modern database management systems, object-oriented or graph-oriented database management systems. The still unresolved challenge of graph-theoretic approaches is the creation, maintenance and update of the graph elements in the case of very large and diverse data and knowledge sources.

The core notion that guided our research presented in this chapter is based on the concept of *bisociation*, as defined by Koestler [11] and refined in our context by Dubitzky et al. [6]. Furthermore, Petrič et al. [15] explore the analogy between Koestler's creativity model and comparable cross-domain knowledge discovery approaches from the field of literature mining. In the field of biomedical literature-mining, Swanson [24] designed the *ABC model* approach, which investigates whether agent *A* is connected with phenomenon *C* by discovering complementary structures via interconnecting phenomena *B*. The process of discovery when domains A and C are known in advance and the goal is to find interconnecting concepts from B is called a *closed discovery process.* On the other hand, if only domain A is known then this is an *open discovery process* since also domain C has to be discovered.

Our research deals only with the closed discovery setting and is to some extent similar to the work of Smalheiser and Swanson [21] where they developed an online system ARROWSMITH, which takes as input two sets of titles from disjoint domains *A* and *C* and lists bridging terms (*b*-terms) that are common to literature *A* and *C*; the resulting b-terms are used to generate novel scientific hypotheses. Other related works in the domain of biomedical literature mining are work of Weeber et al. [28] where authors partly automate Swanson's discovery and work of Srinivasan et al. [23] where they develop an algorithm for bridging term identification with even less expert interaction needed.

This work extensively uses the concepts of bisociation, bridging concept, b-term identification, closed discovery, cross-context and A-C domains presented in the previous paragraph. Furthermore, we have based the evaluation techniques mostly on the results reported by Swanson et al. [26] and Urbančič et al. [27].

The chapter is structured as follows. The second section explains the initial problem we are solving into much more detail, defines the terminology used in this work and outlines the structure of the solution proposed in this chapter. The next section is more technical and it lays ground for some basic procedures for retrieving and preprocessing a collection of documents. It also introduces the standard text-mining procedures and terminology which is essential for understanding the subsequent sections. The fourth section presents the core contribution of this work, i.e., bisociative bridging concept identification techniques which are used to extract key network concepts

(nodes). Evaluation of these core ideas on a previously well studied domain is presented in the following section. The sixth section builds upon the results from concept identification part (Sections 4 and 5) and shows how the final information networks are constructed.

2 Problem Description

This section describes the problem addressed in this work. The initial goal is straightforward: to construct an information network from text documents. The input to the procedure consists of text documents (e.g., titles and abstract of scientific documents) from two disparate domains. The output of the procedure is an information network which could, for example, look like the graph shown in Fig. 1. However, the strong bias towards bisociations leads us to using advanced bridging term identification techniques for detecting important network nodes and relations. The following paragraphs define in detail the input, the output, open issues and sketch the proposed solution.

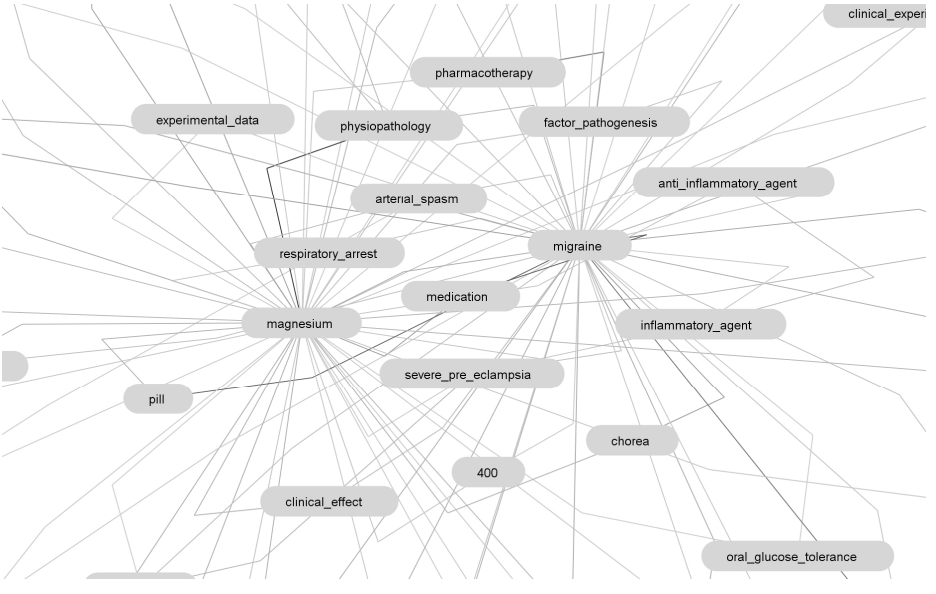

Fig. 1. Part of a network created from PubMed articles on migraine and magnesium

This chapter focuses – similarly as related work from the literature-mining field – on text documents as the primary data source. Texts are in general considered to be one of the most unstructured data sources available, thus, constructing a meaningful graph of data and knowledge (also named an information network) is even more of a challenge.

We are solving the closed discovery problem, which is the topic of research of this chapter and one of the basic assumptions of our methodology. The selected source

text documents are originating from at least two dissimilar domains (M1 and M2 contexts by Koestler's naming or A and C domains according to Swanson and his followers). In this chapter, we always describe the methodology using exactly two domains even though it could be generalised to three or more domains.

In this work, the selected knowledge representation formalism is the so-called *bisociative information network*, called *BisoNet*. The BisoNet representation, as investigated in the BISON[1] project and discussed by Kötter and Berthold [12] is a graph representation, consisting of labeled nodes and edges (see Fig. 1). The original idea underlying the BISON project was to have a node for every relevant concept of an application domain, captured by terms denoting these concepts, that is, by *named entities*. For example, if the application domain is drug discovery, the relevant (named) entities are diseases, genes, proteins, hormones, chemical compounds etc. The nodes representing these entities are connected if there is evidence that they are related in some way. Reasons for connecting two terms/concepts can be linguistic, logical, causal, empirical, a conjecture by a human expert, or a co-occurrence observed in documents dealing with considered domains. E.g., an edge between two nodes may refer to a document (for example, a research paper) that includes the represented entities. Unlike semantic nets and ontologies, a BisoNet carries little semantics and to a large extend encodes just circumstantial evidence that concepts are somehow related through edges with some probability.

Open issues in BisoNet creation are how to identify entities and relationships in data, especially from unstructured data like text documents; i.e., which nodes should be created from text documents, what edges should be created, what are the attributes with which they are endowed and how should element weights be computed. Among a variety of solutions, this chapter presents the one that answers such questions by optimizing the main criterion of generated BisoNets: maximizing their bisociation potential. Bisociation potential is a feature of a network that informally states the probability that the network contains a bisociation. Thus, we want to be able to generate such BisoNets that contain as many bisociations as possible using the given data sources. In other words, maximizing the bisociation potential of the generated BisoNet is our main guidance in developing the methodology for creating BisoNets from text documents.

When creating large BisoNets from texts, we have to address the same two issues as in network creation from any other source: define a procedure for identifying key nodes, and define a procedure for discovering relations among the nodes. However, in practice, a workflow for converting a set of documents into a BisoNet is much more complex than just identifying entities and relations. We have to be able to preprocess text and filter out noise, to generate a large number of entities, evaluate their bisociation potential and effectively calculate various distance measures between the entities. As these tasks are not just conceptually difficult, but also computationally very intensive, great care is needed when designing and implementing algorithms for BisoNet construction.

[1] Bisociation Networks for Creative Information Discovery: `http://www.BisoNet.eu/`

Our approach to confront the network construction problem is based on developing the following ingredients:

1. Provide basic procedures for automatic text acquisition from different sources of interest on the Web.
2. Employ the state of the art approaches for text preprocessing to extract as much information as available in raw text for the needs of succeeding procedures.
3. Incorporate as much as possible available background knowledge into the stages of text preprocessing and candidate concept detection.
4. Define a candidate concept detection method.
5. Develop a method for relevant bisociative concept extraction from identified concept candidates and perform its evaluation.
6. Select a set of relevant extracted bisociative concepts to form the nodes of a BisoNet.
7. Construct relations between nodes and set their weights according to the Bisociation Index measure published and evaluated by Segond and Borgelt [4].

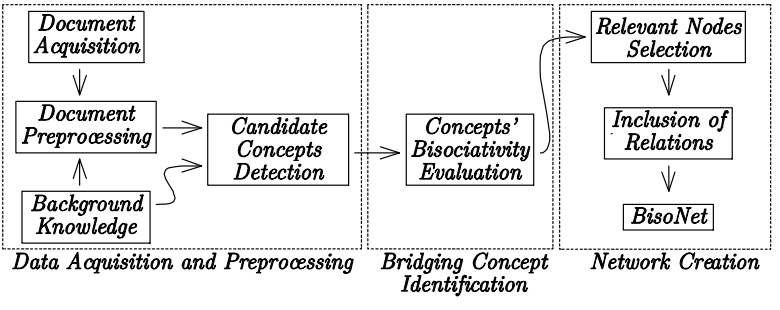

Fig. 2. Conceptual workflow of the proposed solution for BisoNet creation

Fig. 2 illustrates the steps of the methodology proposed by our work. This chapter concentrates mostly on the part of the new methodology for bridging concept evaluation (frame in the middle Fig. 2). As this is an important scientific contribution we provide an evaluation that justifies the design choices in our methodology conception. An evaluation of the final results – BisoNets – is not provided since an experimental evaluation is hard, if not impossible, to construct according to the data we currently possess and work on. We argue that by providing evaluation for high-quality bridging concept identification and evaluation (done in this work) and using the proven bisociative relation measure (defined by Segond and Borgelt [4]), the resulting BisoNets are also of high quality according to the loos defined measure of bisociation potential.

3 Document Acquisition and Preprocessing

This section describes the data preparation part (leftmost frame in Fig. 2) and is written from a technical perspective as it sets grounds for the reproducibility of the subsequent scientifically more interesting steps. Alongside the reproducibility, it addresses

also the introduction of some essential text-mining concepts, which are crucial for understanding specific parts of our methodology. A top-level overview of the methodology, discussed along with a description of the actual working system, defines the preprocessing steps supporting the main goal addressed by this work – bisociative concept detection.

The system for text processing proposed and implemented in this work, named TexAs (Text Assistant), was used to produce the results presented in this chapter. The described TexAs implementation is built on top of the LATINO[2] library (Link analysis and text mining toolbox). This library contains a majority of elementary text mining procedures, but, as the creation of BisoNet is a very specific task (in the field of text mining), a lot of modules had to be implemented from scratch or at least optimized considerably.

3.1 Document Acquisition

For the study, we use only one data source, i.e., PubMed[3], which was used to retrieve the datasets (migraine-magnesium) used in the following sections. However, when experimenting with other domains, we identified and partly supported in TexAs the following text acquisition scenarios:

- Using locally stored files in various application dependent formats – this is the traditional setting in data mining; however, it usually requires large amounts of partly manual work for transforming the data between different formats.
- Acquiring documents using the SOAP web services (e.g. PubMed uses SOAP web service interface to access their database).
- Selecting documents from the SQL databases – it is a fast and efficient but rarely available option.
- Crawling the internet gathering documents from web pages (e.g. Wikipedia).
- Collecting documents from snippets returned from web search engines.

3.2 Document Preprocessing

In addition to explaining various aspects of preprocessing, this section also briefly describes basic text mining concepts and terminology, some of which are taken from the work of Feldman and Sanger [8]. Preprocessing is an important part of network extraction from text documents. Its main task is the transformation of unstructured data from text documents into a predefined well-structured data representation. As shown below, preprocessing is inevitability very tightly connected to the extraction of network entities. In our case, actual bisociative concept candidates are defined already when preprocessing is finished. The subsequent processing step 'only' ranks the entities and to remove the majority of lower ranked entities from the set.

---

[2] LATINO library: `http://sourceforge.net/projects/latino/`

[3] PubMed: A service of U.S. National Library of Medicine, which comprises more than 20 million citations for biomedical literature: `http://www.ncbi.nlm.nih.gov/pubmed`

In general, the task of preprocessing consists of the extraction of *features* from text documents. The set of all features selected for a given document collection is called a *representational model*. Each document is represented by a vector of numerical quantities – one for each aligned feature of the selected representational model. Using this construction, we get the most standard text mining document representation called feature vectors where each numerical component of a vector is related to a feature and represents a form of weight related to the importance of the feature in the selected document. Usually the majority of weights in a vector are equal to zero showing that one of the characteristics of feature vectors is their sparseness – they are often referred to as sparse vectors. The goal of preprocessing is to extract a feature vector for each document from a given document collection.

Commonly used document features are characters, words, terms and concepts [8]. Characters and words carry little semantic information and are therefore not interesting to consider. Terms and concepts on the contrary carry much more semantic information. Terms are usually considered as single or multiword phrases selected from the corpus by means of term-extraction mechanisms (e.g. because of their high frequency) or are present in an external lexicon of a controlled vocabulary. Concepts or keywords are features generated for documents employing the categorization or annotation of documents. Common concepts are derived from manually annotating a document with some predefined keywords or by inserting a document into some predefined hierarchy. When we refer to document features, we mean the terms and the concepts that we were able to extract from the documents. In the rest of this chapter, we do not distinguish between terms or concepts. In the case if a document set contains both, we merge them and pretend that we have only one type of document features, i.e. terms/concepts.

A standard collection of preprocessing techniques [8] is listed below, together with a set of functionalities implemented in our system TexAs:

- *Tokenization*: continuous character stream must be broken up into meaningful sub-tokens, usually words or terms in the case where a controlled vocabulary is present. Our system uses a standard Unicode tokenizer: it mainly follows the Unicode Standard Annex #29 for Unicode Text Segmentation[4]. The alternative is a more advanced tokenizer, which tokenizes strings according to a predefined controlled vocabulary and discards all the other words/terms.
- *Stopword removal*: stopwords are predefined words from a language that usually carry no relevant information (e.g. articles, prepositions, conjunctions etc.); the usual practice is to ignore them when building a feature set. Our implementation uses a predefined list of stopwords – some common lists that are already included in the library are taken from Snowball[5].
- *Stemming or lemmatization*: the process that converts each word/token into the morphologically neutral form. The following alternatives have been made

[4] Unicode Standard Annex #29:
`http://www.unicode.org/reports/tr29/#Word_Boundaries`

[5] Snowball – A small string processing language designed for creating stemming algorithms:
`http://snowball.tartarus.org`

available: Snowball stemmers, the Porter stemmer [17], and the one that we prefer, the LemmaGen lemmatization system [10].
- *Part-of-speech (POS) tagging*: the annotation of words with the appropriate POS tags based on the context in which they appear.
- *Syntactic parsing*: performs a full syntactical analysis of sentences according to a certain grammar. Usually shallow (not full) parsing is used since it can be efficiently applied to large text corpora.
- *Entity extraction*: methods that identify which terms should be promoted to entities and which not. Entity extraction by grouping words into terms using n-gram extraction mechanisms (an n-gram is a sequence of n items from a given sequence) has been implemented in TexAs.

3.3 Background Knowledge

Since high-quality features are hard to acquire, all possible methods that could improve this process should be used at this point. The general approach that usually helps the most consists in incorporating background knowledge about the documents and their domain. The most elegant technique to incorporate background knowledge is to use a controlled vocabulary. A controlled vocabulary is a lexicon of all terms that are relevant in a given domain. Here we can see a major difference when processing general documents as compared to scientific documents. For many scientific domains there exists not only a controlled vocabulary, but also a pre-annotation for a lot of scientific articles. In this case we can quite easily create feature vectors since we have terms as well as concepts already pre-defined. Other interesting approaches to identifying concepts include methods such as KeyGraph [13], which extract terms and concepts with minimal assumptions or background knowledge, even from individual documents. Other alternatives are using domain ontologies which could be, for example, semi-automatically retrieved by a combination of tools such as OntoGen and TermExtractor [9].

3.4 Candidate Concept Detection

The design choice of our approach is that the entities of the BisoNets will be the features of documents, i.e., the terms and concepts defined in the previous section. The subsequent steps are independent of term and concept detection procedure.

Entities need to be represented in a way which enables efficient calculation of different distance measures between the entities. We chose a representation in which an entity is described by a set (vector) of documents in which it appears. In the same way as documents are represented as sparse vectors of features (entities), the entities can also be represented as sparse vectors of documents. This is illustrated in Example 1 where entity ent_1 is present in documents doc_1, doc_3 and doc_4 and hence its feature vector consists of all these documents (with appropriate weights). By analogy to the original vector space – the *feature space* – the newly created vector space is named a *document space*.

Documents	Extracted entities
doc_1	ent_1, ent_2, ent_3
doc_2	ent_3, ent_4, ent_4
doc_3	ent_1, ent_2, ent_2, ent_5
doc_4	ent_1, ent_1, ent_1, ent_3, ent_4, ent_4

Original documents and extracted entities

Feature space	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$w^f_{1:1}$	$w^f_{1:2}$	$w^f_{1:3}$		
doc_2			$w^f_{2:3}$	$w^f_{2:4}$	
doc_3	$w^f_{3:1}$	$w^f_{3:2}$			$w^f_{3:5}$
doc_4	$w^f_{4:1}$		$w^f_{4:3}$	$w^f_{4:4}$	

Sparse matrix of documents: $w^f_{x:y}$ denotes the weight (in the feature space) of entity y in the feature vector of document x

Document space	doc_1	doc_2	doc_3	doc_4
ent_1	$w^d_{1:1}$		$w^d_{1:3}$	$w^d_{1:4}$
ent_2	$w^d_{2:1}$		$w^d_{2:3}$	
ent_3	$w^d_{3:1}$	$w^d_{3:2}$		$w^d_{3:4}$
ent_4		$w^d_{4:2}$		$w^d_{4:4}$
ent_5			$w^d_{5:3}$	

Sparse matrix of entities: $w^d_{x:y}$ denotes the weight (in the document space) of document y in the document vector of entity x

Example 1: Conversion between the feature and the document space

Note that if we write document vectors in the form of a matrix, then the conversion between the feature space and the document space is performed by simply transposing the matrix (see Example 1). The only question that remains open for now is what to do with the weights? Is weight $w^f_{x:y}$ identical to weight $w^d_{y:x}$? This depends on various aspects, but mostly on how we define weights of the entities in the first place when defining document vectors.

There are four most common weighting models for assigning weights to features:

- *Binary*: a feature weight is either one, if the corresponding feature is present in the document, or zero otherwise.
- *Term occurrence*: a feature weight is equal to the number of occurrences of this feature. This weight might be sometimes better than a simple binary since frequently occurring features are likely more relevant as repetitions indicate that the text is strongly concerned with them.
- *Term frequency*: a weight is derived from the term occurrence by dividing the vector by the sum of all vector's weights. The reasoning of the quality of such weight is similar to term occurrence with the additional normalization that equalizes each document importance – regardless of its length.
- *TF-IDF*: Term Frequency-Inverse Document Frequency is the most common scheme for weighting features. It is usually defined as:

$$w^{TFIDF}_{x:y} = \text{TermFreq}\left(ent_x, doc_y\right) \cdot \log(N/DocFreq(ent_x)),$$

where TermFreq(ent_x, doc_y) is the frequency of feature ent_x inside document doc_y (equivalent to term frequency defined in bullet point above), N is the number of all documents and $DocFreq(ent_x)$ is the number of documents that contain ent_x. The idea behind the TF-IDF measure is to lower the weight of features that appear in many documents as this is usually an indication of them being less important (e.g. stopwords). The quality of this approach has also been quantitatively proven by numerous usages in solutions to various problems in text-mining.

These four methods can be further modified by vector normalization (dividing each vector so that the length – usually the Euclidian or Manhattan length – of the vector is 1). If and when this should be done depends on several factors: one of them is the decision which distance measure will be used in the next, the relation construction step. If the cosine similarity is used, a pre-normalization of the vectors is irrelevant, as this is also done during the distance calculation. Example 2 shows the four measures in practice – documents are taken from the first table in Example 1. Weights are calculated for the feature space and are not normalized.

It is worthwhile to note again the analogy between the feature space and the document space. Although we have developed the methodology for entities network extraction, the developed approach can be used also for document network extraction. Moreover, both approaches can be used to extract a unified network representation where documents and entities are nodes, connected using some special relations.

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	1	1	1		
doc_2			1	1	
doc_3	1	1			1
doc_4	1		1	1	

Binary weight

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	1	1	1		
doc_2			1	2	
doc_3	1	2			1
doc_4	3		1	2	

Term occurrence

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$^1/_3$	$^1/_3$	$^1/_3$		
doc_2			$^1/_3$	$^2/_3$	
doc_3	$^1/_4$	$^2/_4$			$^1/_4$
doc_4	$^3/_6$		$^1/_6$	$^2/_6$	

Term frequency

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$(^1/_3) \cdot \log(^4/_3)$	$(^1/_3) \cdot \log(^4/_2)$	$(^1/_3) \cdot \log(^4/_3)$		
doc_2			$(^1/_3) \cdot \log(^4/_3)$	$(^2/_3) \cdot \log(^4/_2)$	
doc_3	$(^1/_4) \cdot \log(^4/_3)$	$(^2/_4) \cdot \log(^4/_2)$			$(^1/_4) \cdot \log(^4/_1)$
doc_4	$(^3/_6) \cdot \log(^4/_3)$		$(^1/_6) \cdot \log(^4/_3)$	$(^2/_6) \cdot \log(^4/_2)$	

TF-IDF: term frequency – inversed document frequency

Example 2: Weighting models of features in document vectors (from Example 1)

3.5 Distance Measures between Vectors

Although distance calculation addressed in this section is not used in the document preprocessing step, it is explained at this point since the content is directly related to the Section 3.4, and since the distance measures are extensively used in the two following sections about bridging concept identification as well as network creation.

The most common measures in vector spaces, which are also implemented in our system TexAs, are the following:

- Dot product: $\mathrm{DotProd}(vec_x, vec_y)$.
- Cosine similarity: $\mathrm{CosSim}(vec_x, vec_y) = \frac{\mathrm{DotProd}(vec_x, vec_y)}{|vec_x| \cdot |vec_y|}$.
  is the dot product normalized by the length of the two vectors. In the cases where the vectors are already normalized, the cosine similarity is identical to the dot product.
- Jaccard index: this similarity coefficient measures the similarity between sample sets. It is defined as the cardinality of the intersection of the sample sets:
  $$\mathrm{JaccInx}(vec_x, vec_y) = \frac{|vec_x \cap vec_y|}{|vec_x \cup vec_y|} = \frac{\mathrm{DotProd}(vec_x, vec_y)}{|vec_x| + |vec_y| - \mathrm{DotProd}(vec_x, vec_y)},$$
  where lengths $|vec_x|$ and $|vec_y|$ are Manhattan lengths of these vectors.
- Bisociation index: it is the similarity measure defined for the purpose of bisociation discovery in the BISON project. It is explained in more detail in [4]. This measure cannot be expressed by the dot product. Therefore, the following definition uses the notation from Example 1:
  $$\mathrm{BisInx}(vec_x, vec_y) = \sum_{i=0}^{M} \left(\sqrt[k]{w_{x:i} \cdot w_{y:i}} \cdot \left(1 - \frac{|\tan^{-1}(w_{x:i}) - \tan^{-1}(w_{y:i})|}{\tan^{-1}(1)}\right)\right),$$
  where M is the number of all the entities.

In general, the choice of a suitable distance measure should be tightly connected to the choice of the weighting model. Some of the combinations are very suitable and have understandable interpretations or were experimentally evaluated as useful, while others are less appropriate. We list the most commonly used pairs of weighting model and distance measure below:

- *TF-IDF weighting and cosine similarity*: this is the standard combination for computing the similarity in the feature space.
- *Binary weighting and dot product distance*: if this is used in the document space the result is the co-occurrence measure, which counts the number of documents where two entities appear together.
- *Term occurrence weighting and dot product distance*: this is another measure of co-occurrence of entities in the same documents. Compared to the previous measure, this one considers also multiple co-occurrences of two entities inside a document and gives them a greater weight in comparison with the case were each appears only once inside the same document.
- *Binary weighting and Jaccard index distance*: Jaccard index was primary defined on sets, therefore the most suitable weighting model to use with it is the binary weighting model (since every vector then represents a set of features).
- *Term frequency weighting and the Bisociation Index distance*: the Bisociation Index was designed with the term frequency weighting in mind, thus it is reasonable to use this combination when determining a weighting model for the Bisociation index.

4 Identifying Bridging Concept Candidates for High Quality Network Entities Extraction

This section presents the key part of our methodology for bisociative bridging terms identification. We propose a set of heuristics which are promising for b-term discovery. In Section 5 we use them to rank all the terms from a document collection and thus obtain some terms which have a higher probability of being b-terms than a randomly selected term.

4.1 Heuristics Description

Heuristics are functions that numerically evaluate the term's quality by assigning a *bisociation score* (tendency that a term is a b-term) to it. For the definition of an appropriate set of heuristics we define a set of special (mainly statistical) properties of terms which will separate b-terms from regular terms. Thus, these heuristics can also be viewed as advanced term statistics.

All heuristics operate on the data retrieved from the documents in preprocessing or obtained from the background knowledge. Using an ideal heuristic and sorting all the terms by the its calculated bisociation scores should result in finding all the b-terms at the top of a list. However, sorting by actual heuristic bisociation scores (either ascending or descending) should still bring much more b-terms than non b-terms to the top of the term list.

Formally, a heuristic is a function with two inputs, i.e., a set of domain labeled documents D and a term t appearing in these documents, and one output, i.e., a number that correlates with the term's bisociation score.

In this chapter we use the following notation: to say that the bisociation score b is equal to the result of a heuristic named $heurX$, we can write it as $b = heurX\ (D, t)$. However, since the set of input documents is static when dealing with a concrete dataset, we can – for the sake of simplicity – omit the set of input documents from a heuristic notation and use only $b = heurX\ (t)$. Whenever we need to explicitly specify the set of documents on which the function works (never needed for a heuristic, but sometimes needed for auxiliary functions used in a formula for a heuristic), we write it as $funcX_D(t)$. For specifying an auxiliary function's document set we have two options: either we use D_u that stands for the (union) set of all the documents from all the domains, or we use $D_n : n \in \{1..N\}$, which stands for a set of documents from the domain n. In general the following statement holds: $D_u = \bigcup_{n=1}^{N} D_n$ where N is the number of domains. In the most common scenario, where we have exactly two distinct domains, we also use the notation D_A for D_1 and D_C for D_2, since we introduced A and C as representatives of the initial and the target domain in the closed discovery setting introduced in Section 1. Due to a large number of heuristics and auxiliary functions we use a multi word naming scheme for easier distinction; names are formed by word concatenation and capitalization of all non-first words (e.g.: *freqProdRel* and *tfidfProduct*).

It is valuable to note that all the designed heuristics are symmetric in the domains, as switching the order of domains (which domain is the initial domain and which is the target) should not affect the outcome of a heuristic. By allowing asymmetric heuristics the approach would lose generality and also the possibility to generalize it to more than two domains.

We divided the heuristics into different sets for easier explanation; however, most of the described heuristics work fundamentally in a similar way – they all manipulate solely the data present in document vectors and derive the terms' bisociation score. The only exceptions to this are the outlier based heuristics which firstly calculate outlier documents and only later use the information from the document vectors.

The heuristics can be logically divided into four sets which are based on: frequency, tf-idf, similarity, and, outliers. Besides those sets we define also two special heuristics which are used as a baseline for other heuristics.

4.2 Frequency Based Heuristics

For easier definition of frequency based heuristics we need two auxiliary sub-functions:

- $countTerm_D(t)$: counts the number of occurrences of term t in a document set D (called term frequency in tf-idf related contexts),
- $countDoc_D(t)$: counts the number of documents in which term t appears in a document set D, (called document frequency in tf-idf related contexts).

We define the following basic heuristics:

(1) $freqTerm(t) = countTerm_{D_u}(t)$: term frequency across both domains,

(2) $freqDoc(t) = countDoc_{D_u}(t)$: document frequency across both domains,

(3) $freqRatio(t) = \frac{countTerm_{D_u}(t)}{countDoc_{D_u}(t)}$: term to document frequency ratio,

(4) $freqDomnRatioMin(t) = \min\left(\frac{countTerm_{D_1}(t)}{countTerm_{D_2}(t)}, \frac{countTerm_{D_2}(t)}{countTerm_{D_1}(t)}\right)$: minimum of term frequencies ratio between both domains,

(5) $freqDomnProd(t) = countTerm_{D_1}(t) \cdot countTerm_{D_2}(t)$: product of term frequencies in both domains,

(6) $freqDomnProdRel(t) = \frac{countTerm_{D_1}(t) \cdot fcountTerm_{D_2}(t)}{countTerm_{D_u}(t)}$: product of term frequencies in both domains relative to the term frequency in all domains.

4.3 Tf-idf Based Heuristics

Tf-idf is the standard measure of term's importance in a document which is used heavily in text mining research. In the following heuristic definitions we use the following auxiliary functions:

- $tfidf_d(t)$ stands for tf-idf of a term t in a document d, and,
- $tfidf_D(t)$ represents tf-idf of a term in the centroid vector of all the documents $d: d \in D$. The centroid vector is defined as an average of all document vectors and thus presents an average document from the document collection D

Heuristics based on tf-idf are listed below:

(7) $tfidfSum(t) = \sum_{d \in D_u} tfidf_d(t)$: sum of all tf-idf weights of a term across both domains – analogy to $freqTerm(t)$,

(8) $tfidfAvg(t) = \frac{\sum_{d \in D_u} tfidf_d(t)}{freq\_doc_{D_u}(t)}$: average tf-idf of a term,

(9) $tfidfDomnProd(t) = tfidf_{D_1}(t) \cdot tfidf_{D_2}(t)$: product of a term's importance in both domains.

(10) $tfidfDomnSum(t) = tfidf_{D_1}(t) + tfidf_{D_2}(t)$: sum of a term's importance in both domains.

4.4 Similarity Based Heuristics

Another approach to construct a relevant heuristic measures is to use the cosine similarity measure. We start by creating a representational model as a document space and by converting terms (entities) into document vectors (see section 3.4). Next, we get the centroid vectors for both domains in the document space representation. Furthermore, we apply tf-idf weighting on top of all the newly constructed vectors and centroids. Finally we use the following auxiliary function to construct the heuristics:

— $simCos_D(t)$: calculates the cosine similarity of the document vector of term t and the document vector of a centroid of documents $d \in D$.

Constructed heuristics:

(11) $simAvgTerm(t) = simCos_{D_u}(t)$: similarity to an average term – the distance from the center of the cluster of all terms,

(12) $simDomnProd(t) = simCos_{D_1}(t) \cdot simCos_{D_2}(t)$: product of a term's similarity to the centroids of both domains,

(13) $simDomnRatioMin(t) = \min\left(\frac{simCos_{D_1}(t)}{simCos_{D_2}(t)}, \frac{simCos_{D_2}(t)}{simCos_{D_1}(t)}\right)$: minimum of a term's frequencies ratio between both domains.

4.5 Outlier Based Heuristics

Conceptually, an outlier is an unexpected event, entity or – in our case – document. We are especially interested in outlier documents since they frequently embody new information that is often hard to explain in the context of existing knowledge. Moreover, in data mining, an outlier is frequently a primary object of study as it can potentially lead to the discovery of new knowledge. These assumptions are well aligned with the bisociation potential that we are optimizing, thus, we have constructed a couple of heuristics that harvest the information possibly residing in outlier documents.

We concentrate on a specific type of outliers, i.e., domain outliers, which are the documents that tend to be more similar to the documents of the opposite domain than to those of their own domain. The procedures that we use to detect outlier documents build a classification model for each domain and afterwards classify all the documents using the trained classifier. The documents that are misclassified are declared as outlier documents, since according to the classification model they do not belong to their domain of origin.

We defined three different outlier sets based on three classification models used. These outlier sets are:

- D_{CS}: retrieved by Centroid Similarity (CS) classifier,
- D_{RF}: retrieved by Random Forest (RF) classifier,
- D_{SVM}: retrieved by Support Vector Machine (SVM) classifier.

Centroid similarity is a basic classifier model and is also implemented in the TexAs system. It classifies each document to the domain whose centroid's tf-idf vector is the most similar to the document's tf-idf vector. The description of the other two classification models is beyond the scope of this chapter, as we used external procedures to retrieve these outlier document sets. The detailed description is provided by Sluban et al. [20].

For each outlier set we defined two heuristics: the first counts the frequency of a term in an outlier set and the second computes the relative frequency of a term in an outlier set compared to the relative frequency of a term in the whole dataset. The resulting heuristics are listed below:

(14) $outFreqCS(t) = countTerm_{D_{CS}}(t)$: term frequency in CS outlier set,

(15) $outFreqRF(t) = countTerm_{D_{RF}}(t)$: term frequency in RF outlier set,

(16) $outFreqSVM(t) = countTerm_{D_{SVM}}(t)$: term frequency in SVM outlier set,

(17) $outFreqSum(t) = countTerm_{D_{CS}}(t) + countTerm_{D_{RF}}(t) + countTerm_{D_{SVM}}(t)$: sum of term frequencies in all three outlier sets,

(18) $outFreqRelCS(t) = \frac{countTerm_{D_{CS}}(t)}{countTerm_{D_u}(t)}$: relative frequency in CS outlier set,

(19) $outFreqRelRF(t) = \frac{countTerm_{D_{RF}}(t)}{countTerm_{D_u}(t)}$: relative frequency in RF outlier set,

(20) $outFreqRelSVM(t) = \frac{countTerm_{D_{SVM}}(t)}{countTerm_{D_u}(t)}$: relative frequency in SVM outlier set,

(21) $outFreqRelSum(t) = \frac{countTerm_{D_{CS}}(t) + countTerm_{D_{RF}}(t) + countTerm_{D_{SVM}}(t)}{countTerm_{D_u}(t)}$: sum of relative term frequencies in all three outlier sets.

4.6 Baseline Heuristics

We have two other heuristics which are supplementary and serve as a baseline for the others. The auxiliary functions used in their calculation are:

- $randNum()$: returns random number from the interval (0,1) regardless of the term under investigation,
- $inBoth(t)$: 1 if a term t appears in both domains and 0 otherwise.

The two baseline heuristics are:

(22) $random(t) = randNum()$: random baseline heuristic,

(23) $appearInAllDomn(t) = inBoth(t) + (randNum())/2$: it is a better baseline heuristic which can separate two classes of terms – the ones that appear in both domains and the ones that appear only in one. The terms that appear only in one domain have a strictly lower heuristic score than those that appear in both. The score inside of these two classes is still random.

5 Heuristics Evaluation

This section presents the evaluation of the heuristics defined in the previous section. First we describe the evaluation procedure, then the domain on which we evaluate the heuristics is presented, and finally the results of the evaluation along with the discussion of the results.

5.1 Evaluation Procedure

In the experimental setting used in this chapter we are given the following: a set of documents from two domains and a "gold standard" list of b-terms. Consequently, we are able to mark the true b-terms and evaluate how well our constructed heuristics are able to promote these b-terms compared to the rest of the terms.

We compare the heuristics using ROC (Receiver Operating Characteristic) curve and AUC (Area Under ROC) analysis. Some ideas on using the ROC for our evaluation were taken from Foster et al. [18]. ROC curves are constructed in the following way:

- Sort all the terms by their descending heuristic score.
- Starting from the beginning of the term list, do the following for each term: if a term is a b-term, then draw one vertical line segment (up) on the ROC curve, else draw one horizontal line segment (right) on the ROC curve.
- Sometimes, a heuristic outputs the same score for many terms and therefore we cannot sort them uniquely. Among terms with the same bisociation score b, let b_b be the number of terms that are b-terms and nb_b the number of non-b-terms. We then draw a line from the current point p to the point $p + (\mathrm{nb}_b, \mathrm{b}_b)$. In this way we may produce slanted lines, if such an equal scoring term set contains both b-terms and non b-terms.

Using the stated procedure, we get one ROC curve for each heuristic. The ROC space is defined by its two axes. The ROC's vertical axis scale goes from zero to the number of b-terms and the horizontal goes from zero to the number of non b-terms. AUC is defined as the percentage of the area under curve – the area under the curve is divided by the area of the whole ROC space. If a heuristic is perfect (it detects all the b-terms and ranks them at the top of the ordered list), we get a curve that goes first just up and then just right with an AUC of 100%. The worst possible heuristic sorts all the terms randomly regardless of being a b-term or not and achieves AUC of 50%. This random heuristic is represented by the diagonal in the ROC space.

The fact that some heuristics output the same score for many terms can produce different sorted lists and thus different performance estimates for the same heuristic on the same dataset. In the case of such equal scoring term sets, the inner sorting is random (which indeed produces different performance estimates). However, the ROCs that are provided (and constructed by the instructions in the paragraph above) correspond to the average ROC over all possible such random inner sortings. Besides AUC, we list also the interval of AUC which tells how much each heuristic varies among the best and the worst sorting of a possibly existing equal scoring term set. Preferable are the heuristics with a smaller interval which implies that they produce smaller and fewer equal scoring sets.

5.2 Migraine-Magnesium Dataset

This section describes the dataset used to evaluate the heuristics' potential of successful b-term identification. The dataset that we used is the well-researched *migraine-magnesium* domain pair which was introduced by Swanson [24] and later explored by several authors in several studies [25, 28, 26, 14]. In the literature-based discovery process Swanson managed to find more than 60 pairs of articles connecting the migraine domain with the magnesium deficiency via 43 b-terms. In our evaluation we are trying to rediscover these b-terms stated by Swanson to connect the two domains (see Table 1).

Table 1. B-terms identified by Swanson et al. in [26]

1	5 ht	16	convulsive	31	prostaglandin
2	5 hydroxytryptamine	17	coronary spasm	32	prostaglandin e1
3	5 hydroxytryptamine receptor	18	cortical spread depression	33	prostaglandin synthesis
4	anti aggregation	19	diltiazem	34	reactivity
5	anti inflammatory	20	epilepsy	35	seizure
6	anticonvulsant	21	epileptic	36	serotonin
7	antimigraine	22	epileptiform	37	spasm
8	arterial spasm	23	hypoxia	38	spread
9	brain serotonin	24	indomethacin	39	spread depression
10	calcium antagonist	25	inflammatory	40	stress
11	calcium blocker	26	nifedipine	41	substance p
12	calcium channel	27	paroxysmal	42	vasospasm
13	calcium channel blocker	28	platelet aggregation	43	verapamil
14	cerebral vasospasm	29	platelet function		
15	convulsion	30	prostacyclin		

The dataset contains scientific paper titles which were retrieved by querying the PubMed database with the keyword *"migraine"* for the migraine domain and with the keyword *"magnesium"* for the magnesium domain. Additional condition to the query was the publishing date which was limited to before the year 1988, since Swanson's original experiment – which we want to reproduce – also considered only articles published before that year. The query resulted in 8,058 titles (2,425 from the migraine domain and 5,633 from the magnesium domain) of the average length of 11 words. We preprocessed the dataset using the standard procedures described in Section 3.2 and by additionally specifying terms as n-grams of maximum length 3 (max. three words were combined to form a term) with minimum occurrence 2 (each n-gram had to appear at least twice to be promoted to a term). Using this preferences we produced a dataset containing 13,525 distinct terms or 1,847 distinct terms that appear at least once in each domain; both numbers include also all the 43 terms that Swanson marked as b-terms. An average document in the dataset consists of 12 terms and 394 (4,89%) documents contain at least one b-term.

5.3 Comparison of the Heuristics

This section presents the results of the comparison of the heuristics on the magnesium-migraine dataset using ROC analysis. The experimental setting was presented in detail in the previous sections. Nevertheless, for the purpose of this evaluation, it was slightly extended, due to additional knowledge about b-terms in this domain (this may be a general observation for any future domain). We realized that all the 43 b-terms appear in both domains; therefore, it is more fair for the comparison that the heuristics are also aware of this fact. Therefore, we made sure that every heuristic ordered all the terms that appear in both datasets (1,847 terms) before all the other terms (11,678 terms), however, every heuristic used its own score for ordering within these two sets of terms. In this way, we incorporated the stated background knowledge about b-terms in this domain into all the heuristics.

Table 2. Comparison of the results of all the defined heuristics ordered by the quality – AUC. The first column states the name of the heuristic; the second displays a percentage of the area under the ROC curve; and the last is the nterval of AUC.

Heuristic	AUC	Interval
(21) *outFreqRelSum*	95,33%	0,35%
(19) outFreqRelRF	95,24%	0,55%
(20) outFreqRelSVM	95,06%	1,26%
(18) outFreqRelCS	94,96%	1,30%
(17) outFreqSum	94,96%	0,70%
(8) tfidfAvg	94,87%	0,00%
(15) outFreqRF	94,73%	1,53%
(16) outFreqSVM	94,70%	2,06%
(14) outFreqCS	94,67%	1,80%
(4) freqDomnRatioMin	94,36%	0,62%
(10) tfidfDomnSum	93,85%	0,35%
(6) freqDomnProdRel	93,71%	0,40%
(13) simDomnRatioMin	93,58%	0,00%
(7) tfidfSum	93,58%	0,00%
(9) tfidfDomnProd	93,47%	0,39%
(5) freqDomnProd	93,42%	0,44%
(3) freqRatio	93,35%	5,23%
(23) *appearInAllDomn*	93,31%	6,69%
(12) simDomnProd	93,27%	0,00%
(1) freqTerm	93,20%	0,50%
(2) freqDoc	93,19%	0,50%
(11) simAvgTerm	92,71%	0,00%
(22) *random*	50,00%	50,00%

The first look at numerical result comparison (Table 2) reveals the following:

- The overall AUC results of all heuristics, except for the (22)random baseline, are relatively good and in the range of from approx. 93% to 95%.
- The difference among AUC results is small (only 2.5% between the worst and the best performing heuristic).
- The improved baseline heuristic (23)appearInAllDomn performs well and is not worse than some other heuristics.
- Outlier based heuristics seem to perform the best.
- Some heuristics, including the best performing ones, have a relatively high AUC interval which means that they output the same score for many terms.

Observing the results in Table 2, followed by the detailed ROC analysis described below, we selected the best heuristic that will be used as the heuristic for network node weighting, which is the final result of this work. The chosen heuristic is simply the first from the list in Table 2 – (21)outFreqRelSum – due to the fact that it has highest AUC and especially since it shows a low uncertainty. In other words, it has

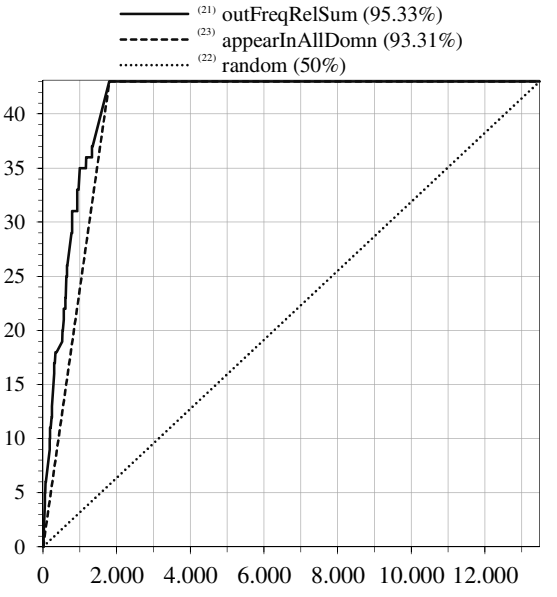

Fig. 3. ROC curve of the selected heuristic (21)outFreqRelSum along with the baseline heuristic (22)random and improved baseline heuristic (23)appearInAllDomn on detecting the 43 b-terms among all 13,525 candidate concepts.

small AUC interval, which means that it better defines the position of b-terms and we do not need to rely so much on random sorting of potential equal scoring term sets. We also assume it to be less volatile across domains since it actually represents cooperation (sum) of three other well performing heuristics: (19)outFreqRelRF, (20)outFreqRelSVM, and, (18)outFreqRelCS.

Detailed ROC curve analysis of the chosen heuristic (see Fig. 3) shows that our heuristic is only slightly better than the improved baseline heuristic, which is evident also from Table 2. However, when examined carefully we perceive the property of the heuristic which is the initial assumption of this research, i.e., extremely steep incline at the beginning of the curve which is much steeper than the incline of the baseline heuristics. This means that the chosen heuristic is able to detect b-terms at the beginning of the ordered list much faster than the baseline. The steep incline is even more evident in Fig. 4.

Fig. 4 shows the zoom-in perspective on the ROC curves of the selected outlier based heuristics – enumerated from (18) to (21) – along with the baselines. The zoom-in (applied also in Fig. 5) refers to the axis x since we show only 1,804 terms which is the point where all the heuristics (except (22)random) reach the top point (43 found b-terms). In Fig. 4 we can see the steep incline property of the (21)outFreqRelSum even more clearly. At the position of the first tick on the axis x (by the term 50 in the ordered list of terms) the chosen heuristic is able to detect already 5-6 b-terms while the baseline heuristic only approximately one. Similarly, we notice at the 200th term the baseline heuristics detects 5 b-terms while (21)outFreqRelSum detects already 11.

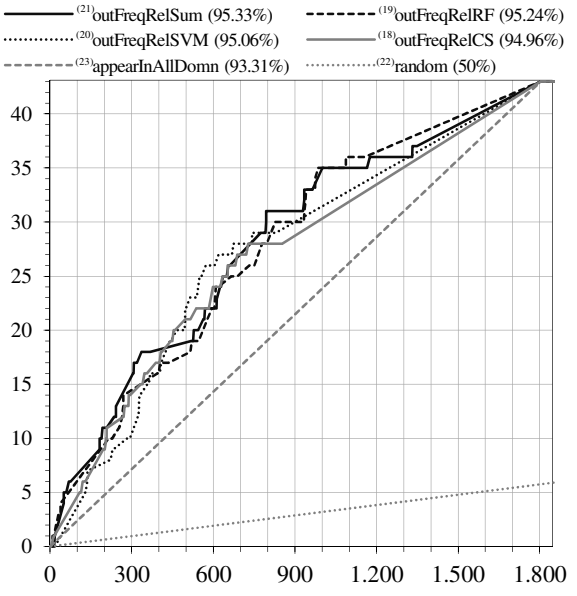

Fig. 4. ROC curves of the best-performing set of heuristic – relative frequency of a term in outlier sets – along with both baseline heuristics on detecting the b-terms among only 1,847 candidate concepts (only the concepts that appear in both domains)

If we follow the curve further we see a decrease in relative difference; nevertheless, at the 1000th term the ratio is still 24:35, even though the performance here is not of such importance as the performance at the beginning of the curve. The presented behavior at the beginning of the curve is highly appreciated especially from the expert's point of view who needs to go through such an ordered list of terms and detect potential b-terms. In such a setting we would really want to present some valuable b-terms at the very beginning of the list, even if other b-terms are dispersed evenly across it.

Even though we chose the heuristic from the outlier set we are still interested how the heuristics from the other sets performed. This comparison is presented in Fig. 5 where we show one (the best performing one) heuristic from each set of heuristics. Notice the outlier heuristic (19)outFreqRelRF which undoubtedly wins. It is harder to establish an order between the other three heuristics. The undesired property is exposed by (13)simDomnRatioMin where the ROC curve shows performance worse than (23)appearInAllDomn at the right side of the curve; however, even this would be tolerable if there is outperformance at the beginning of the curve. The conclusion for the other sets (besides the outlier one) is that even though they are slightly better than the baseline heuristic we are not able to infer their significant outperformance over it.

Overall, the results of the evaluation are beneficial for the insight into heuristic performance on the examined migraine-magnesium dataset. The conclusion is that it is extremely hard to promote b-terms in an ordered list of terms by observing only the terms' statistical properties in the documents. However, we managed to construct a well performing heuristic which is based on relative frequency of a term in three outlier sets of all the documents. The outlier sets of documents are retrieved using

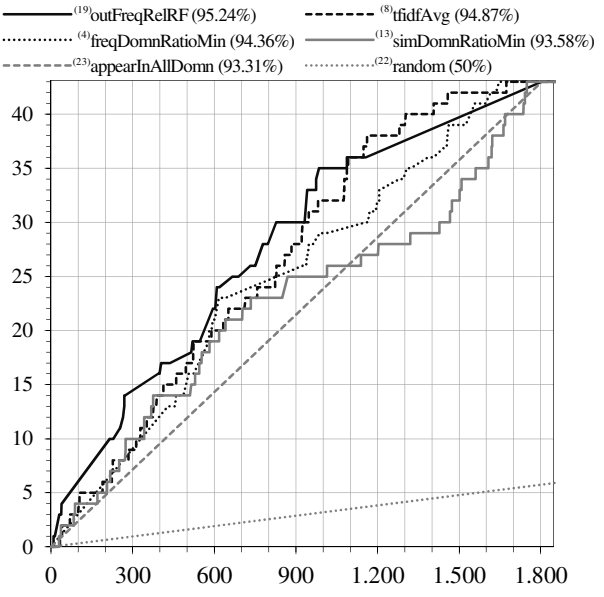

Fig. 5. ROC curves of the best-performing heuristics – one from each set (based on: frequency, tf-idf, similarity, outliers) along with both baseline heuristics on detecting the b-terms among only 1,847 candidate

three types of classifiers: Centroid Similarity, Random Forest, and, Support Vector Machine. The conclusion of our evaluation is well aligned with the results presented by Sluban et al. [20] and Petrič et al. [16].

The presented chapter motivated our future work in several directions of which we will first proceed with the following:

- Reevaluate the findings on a new independent test domains. We have already done some initial tests on the autism-calcineurin domain pair presented by Urbančič et al. [27], which show similar results as the presented evaluation.
- Try to do some further research on heuristics based on statistical properties of the terms. If no heuristics which outperform (23)appearInAllDomn is found, we will consider completely abandoning this type of heuristics.
- Add some new, fundamentally different classes of heuristics to rank the terms. We have a couple of ideas to try, including using SVM keywords (SVM trained to separate between domains) as potential b-terms with high score.
- Implement the findings of this research as a web application where the user (a domain expert) will be able to perform an experimentation and b-term retrieval on his own domains of interest.

6 Network Creation

This section briefly presents the ideas behind the creation of a BisoNet – an information network of concepts identified and weighted by the presented methodology.

The initial plan for BisoNet construction is first to take all the terms/concepts identified in the preprocessing step, next to weight them using the bisociation score of the $^{(21)}$outFreqRelSum heuristic and finally to add links among concepts according to the Bisociation Index measure defined by Segond and Borgelt [4].

Table 3. The 40 highest ranked terms using the preferred heuristic $^{(21)}$outFreqRelSum along with the weights (bisociation score) retrieved by the same heuristic. There are 5 gold standard b-terms in this list and they are all marked with asterisks.

1	sturge	3.50	26	cerebral artery	2.50
2	sturge weber	3.50	27	medication	2.50
3	weber	3.50	28	animal human	2.50
4	inflammatory agent	3.00	29	trial treatment	2.50
5	double blind clinical	3.00	30	*brain serotonin* *	2.50
6	migraine therapy magnesium	3.00	31	comparative double blind	2.50
7	ophthalmologic	3.00	32	comparative double	2.50
8	clinical aspect therapy	3.00	33	400	2.50
9	anti inflammatory agent	3.00	34	hyperventilation	2.50
10	therapy magnesium glutamate	3.00	35	cortical spread	2.50
11	bruxism	3.00	36	concentration serotonin	2.50
12	magnesium glutamate	3.00	37	pill	2.50
13	blind clinical	3.00	38	physiopathological	2.50
14	aspect therapy	3.00	39	vasospastic	2.50
15	physiopathology	2.83	40	respiratory arrest	2.50
16	hypotension	2.66	41	peripheral artery	2.50
17	treatment spontaneous	2.66	42	*spread depression* *	2.43
18	oral glucose tolerance	2.50	43	pharmacotherapy	2.33
19	*cerebral vasospasm* *	2.50	44	*arterial spasm* *	2.33
20	response serum	2.50	45	acid metabolism	2.33
21	factor pathogenesis	2.50	46	clinical experimental study	2.33
22	*cortical spread depression* *	2.50	47	chorea	2.33
23	severe pre	2.50	48	lactase	2.33
24	severe pre eclampsia	2.50	49	arginine	2.33
25	experimental data	2.50	50	clinical effect	2.33

We will explain BisoNet construction by creating an example network from the migraine-magnesium domain pair. Table 3 states first 50 terms which are the output of the first two steps of the procedure: candidate concept detection and $^{(21)}$outFreqRelSum heuristic scoring. How many terms do we consider for inclusion in the final BisoNet depends on the use-case of the created network. In the case when the network is an input of the following automatic procedures for bisociation detection, we want to keep as many nodes as possible, i.e., all candidate concepts nodes (13,525 in the migraine-magnesium domain). There may be a need to trim the number of nodes down either due to the computational complexity of the subsequent bisociation discovery procedures or due to the fact that the network is meant to be explored by a human. In such a case we have two primary options to consider: the first is to remove all the nodes that do not appear in both domains since those are less probable to contain interesting bisociations (we are left with 1,847 nodes in the migraine-magnesium domain). The second option is to use the scores of $^{(21)}$outFreqRelSum to cut the nodes under the specified threshold limit.

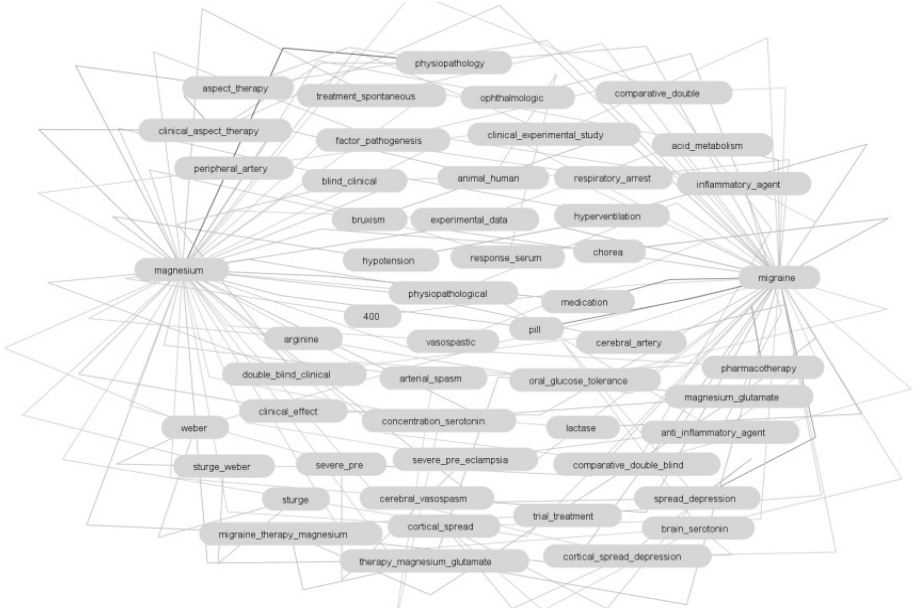

Fig. 6. Part of the network constructed from the migraine-magnesium database using [21]outFreqRelSum heuristic for weighting the nodes and Bisociation Index for weighting the links

The only step remaining to finalize a BisoNet construction is to calculate the links. If we have a reasonably large number of nodes (e.g. 1,000 or more) then it is infeasible to calculate all the links since there are $(n \cdot (n-1))/2$ of them if n is the number of nodes. Therefore, we again use thresholding to cut away lower weighted links. In extreme cases where there is a really vast number of nodes (e.g. 100,000 or more) there are special approaches needed to calculate all the links – even before thresholding is applied and the nodes are stored. However, these algorithms are beyond the scope of this work.

Fig. 6 shows a section of the final BisoNet constructed by the methodology described in this work. A section contains all the highest-ranking nodes retrieved using a threshold on the concepts' [21]outFreqRelSum heuristic score (see Table 3) and the two – in this domain – special nodes: migraine and magnesium. The links among nodes were calculated as described and were not thresholded. Weights on the links and nodes are not shown due to clarity; however, the node weights are stated in Table 3 while link weights can be inferred from the strength – darkness of the links.

With the presentation of this example we conclude this chapter. We addressed the problem of producing an information network, named BisoNet, from a large text corpus consisting of at least two diverse domains. The goal was to produce a BisoNet that has a high potential for providing yet unexplored cross-domain links which could lead to new scientific discoveries. We devoted most of this chapter to the sub-problem: *how to better identify important domain-bridging concepts* which become core nodes of the resulting network. We also provided a detailed description of

all the preprocessing steps required to reproduce this work. The evaluation of bridging concept identification was performed by repeating a discovery made on medical articles in the migraine-magnesium domain. Further work is tightly related to the main focus of this chapter – heuristics for b-term identification and their evaluation – therefore, we stated the ideas for further work at the end of Section 5.

Acknowledgements. The work presented in this chapter was supported by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open project BISON-211898, and the Slovenian Research Agency grant Knowledge Technologies (P2-0103)

References

1. Albert, R., Barabasi, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. 74(1), 47–97 (2002)
2. Bales, M.E., Johnson, S.B.: Graph theoretic modeling of large scale semantic networks. Journal of Biomedical Informatics 39(4), 451–464 (2006)
3. Berthold, M.R., Dill, F., Kötter, T., Thiel, K.: Supporting Creativity: Towards Associative Discovery of New Insights. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 14–25. Springer, Heidelberg (2008)
4. Segond, M., Borgelt, C.: "BisoNet" Generation using Textual Data. In: Proceedings of Workshop on Explorative Analytics of Information Networks at ECML PKDD (2009)
5. Boström, H., Andler, S.F., Brohede, M., Johansson, R., Karlsson, A., van Laere, J., Niklasson, L., Nilsson, M., Persson, A., Ziemke, T.: On the definition of information fusion as a field of research. Technical report, University of Skovde, School of Hum.and Inf., Skovde, Sweden (2007)
6. Dubitzky, W., Kötter, T., Schmidt, O., Berthold, M.R.: Towards Creative Information Exploration Based on Koestler's Concept of Bisociation. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 11–32. Springer, Heidelberg (2012)
7. Dura, E., Gawronska, B., Olsson, B., Erlendsson, B.: Towards Information Fusion in Pathway Evaluation: Encoding Relations in Biomedical Texts. In: Proceedings of the 9th International Conference on Information Fusion (2006)
8. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press (2007)
9. Fortuna, B., Lavrač, N., Velardi, P.: Advancing Topic Ontology Learning through Term Extraction. In: Ho, T.-B., Zhou, Z.-H. (eds.) PRICAI 2008. LNCS (LNAI), vol. 5351, pp. 626–635. Springer, Heidelberg (2008)
10. Juršič, M., Mozetič, I., Lavrač., N.: Learning Ripple Down Rules for Efficient Lemmatization. In: Proceedings of the 10th International Multiconference Information Society 2007, vol. A, pp. 206–209 (2007)
11. Koestler, A.: The Act of Creation. The Macmillan Co. (1964)

12. Kötter, T., Berthold, M.R.: From Information Networks to Bisociative Information Networks. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 33–50. Springer, Heidelberg (2012)
13. Ohsawa, Y., Benson, N.E., Yachida, M.: KeyGraph: Automatic Indexing by Co occurrence Graph based on Building Construction Metaphor. In: Proceedings of the Advances in Digital Libraries Conference (ADL), pp. 12–18 (1998)
14. Petric, I., Urbancic, T., Cestnik, B., Macedoni Luksic, M.: Literature mining method RaJoLink for uncovering relations between biomedical concepts. Journal of Biomedical Informatics 42(2), 219–227 (2009)
15. Petrič, I., Cestnik, B., Lavrač, N., Urbančič, T.: Outlier Detection in Cross Context Link Discovery for Creative Literature Mining. Comput. J., November 2 (2010)
16. Petrič, I., Cestnik, B., Lavrač, N., Urbančič, T.: Bisociative Knowledge Discovery by Literature Outlier Detection. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 313–324. Springer, Heidelberg (2012)
17. Porter, M.F.: An algorithm for suffix stripping. Progr. 14(3), 130–137 (1980)
18. Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning 42(3), 203–231 (2001)
19. Racunas, S., Griffin, C.: Logical data fusion for biological hypothesis evaluation. In: Proceedings of the 8th International Conference on Information Fusion (2005)
20. Sluban, B., Juršič, M., Cestnik, B., Lavrač, N.: Exploring the Power of Outliers for Cross-domain Literature Mining. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 325–337. Springer, Heidelberg (2012)
21. Smalheiser, N.R., Swanson, D.R.: Using ARROWSMITH: a computer assisted approach to formulating and assessing scientific hypotheses. Comput Methods Programs Biomed. 57(3), 149–153 (1998)
22. Smirnov, A., Pashkin, M., Shilov, N., Levashova, T., Krizhanovsky, A.: Intelligent Support for Distributed Operational Decision Making. In: Proceedings of the 9th International Conference on Information Fusion (2006)
23. Srinivasan, P., Libbus, B., Sehgal, A.K.: Mining MEDLINE: Postulating a beneficial role for curcumin longa in retinal diseases. In: Hirschman, L., Pustejovsky, J. (eds.) BioLINK 2004: Linking Biological Literature, Ontologies, and Databases, Boston, Massachusetts, pp. 33–40 (2004)
24. Swanson, D.R.: Migraine and magnesium: Eleven neglected connections. Perspectives in Biology and Medicine 31(4), 526–557 (1988)
25. Swanson, D.R.: Medical literature as a potential source of new knowledge. Bull. Med. Libr. Assoc. 78(1), 29–37 (1990)
26. Swanson, D.R., Smalheiser, N.R., Torvik, V.I.: Ranking Indirect Connections in Literature Based Discovery: The Role of Medical Subject Headings (MeSH). Journal of the American Society for Inf. Science and Technology 57, 1427–1439 (2006)
27. Urbančič, T., Petrič, I., Cestnik, B., Macedoni-Lukšič, M.: Literature Mining: Towards Better Understanding of Autism. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) AIME 2007. LNCS (LNAI), vol. 4594, pp. 217–226. Springer, Heidelberg (2007)
28. Weeber, M., Vos, R., Klein, H., de Jong van den Berg, L.T.W.: Using concepts in literature based discovery: Simulating Swanson's Raynaud–fish oil and migraine–magnesium discoveries. J. Am. Soc. Inf. Sci. Tech. 52(7), 548–557 (2001)